

Methodische Konzepte zur Überwindung kognitiver Barrieren im Informatikunterricht

Michael Weigend
Universität Potsdam
Oktober 2003

Übersicht

1 Kognitive Barrieren im Informatikunterricht

Was sind kognitive Barrieren?

Exkurs: Mentale Konzepte

Fehlvorstellungen

Fehlende Konzepte

Fehlende Lernstrategien

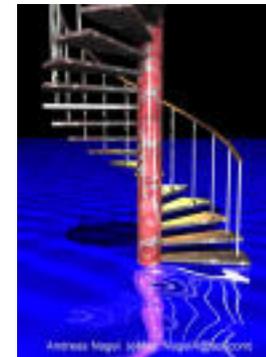
Motivation

2 Erforschung kognitiver Barrieren

3 Methodische Konzepte

4 Was ist zu tun?

1.1 Was sind kognitive Barrieren?



Kognitive Barrieren:

Hindernisse durch fehlendes Wissen

Lewin: Barriere = Element des Lebensraumes

soziales Phänomen

Steuerungsmechanismus in sozialen Systemen

- Schutz vor unberechtigtem Zugang
antike Schlösser, Kryptographie, Steganographie
- Geheimwissen und Geheimsprachen von Gruppen
Expertensprache, Hacker-Jargon

psychisches Phänomen

Hindernis, das Handlungsfreiheit einschränkt und Dazulernen blockiert

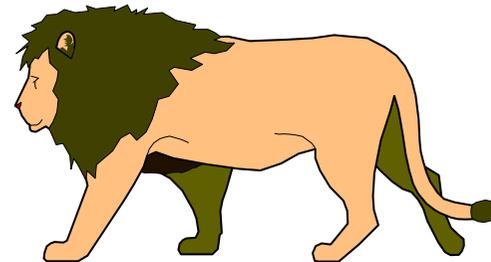
- Fehlvorstellungen (misconceptions)
- Fehlende geeignete Konzepte
- Fixierung auf ungeeignete Konzepte
- Fehlende Lernstrategien
- Fehlende Motivation

Exkurs: Wissensrepräsentation



Begriff

Katze



- bezeichnet Klasse von Objekten und Ereignissen
- besitzt definierende Merkmale

Katze :

{atmet, kann sich bewegen, frisst Fleisch ... }

Schema

Anderson (1996)

Haus

Attribute
(Slots)

Oberbegriff : Gebäude

Teile: Zimmer

Material: Holz, Stein

Funktion: Wohnraum

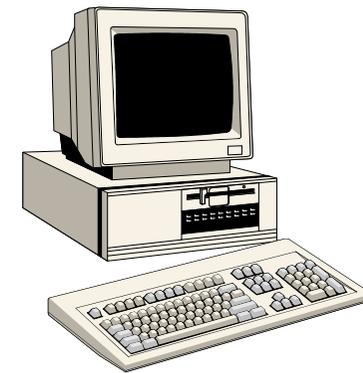
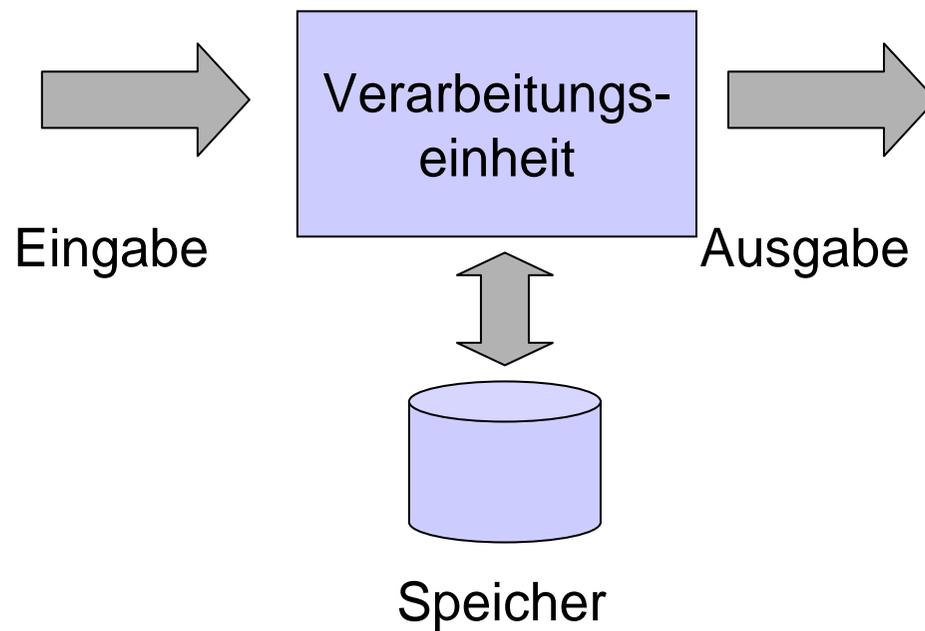
Größe: 10 m² bis 1000 m²

Beziehung
zu anderen
Schemata (ist,hat)

mögliche
Werte

definiert vorhersagbare Informationen
über Exemplare der Kategorie

mentales Modell



Vorstellungswelt
(mind)

reale Welt

1. 2 Fehlvorstellungen

fallender Wassertropfen



Atome und Moleküle

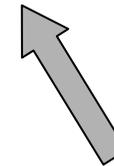
Fehlvorstellung	Natur-Wissenschaftlich	akademisch	Nicht akademisch
Atome leben	4	7	5
Wassermoleküle in einer Stoffprobe haben unterschiedliche Größen	4	10	9
Wassermoleküle in der Gasphase sind die leichtesten	2	6	7

30 Highschool-Schüler/innen
im Alter von 16 bis 18

Griffiths und Preston 1992

Ursachen

Fehlvorstellung



Ursachen

Eigene Beobachtungen
im Alltag

Einzeller unter dem
Mikroskop
herumirrende Asseln

Darstellungen in
Medien

Bücher
Fernsehen
Schulunterricht

Informatische Fehlvorstellungen

Fehlvorstellung	Potentielle Ursachen	Typische Fehler
Eine Anweisung ist das gleiche wie eine Aussage	Alltäglicher Sprachgebrauch „Du nimmst jetzt deine Kappe ab“	Verwechslung von Zuweisung und Vergleich $X = 1$ $X == 1$
Computer können intuitiv agieren (Nichtdeterminismus, Animismus)	Alltagsalgorithmen lassen Entscheidungsspielraum	Gift = 'Cadmium' OR 'Blei'

Umgang mit Fehlvorstellungen in den Naturwissenschaften

- Diagnose

bewusst machen

- Konfrontation

mit Beobachtungen vergleichen
durch Experimente widerlegen

- Heilung

verlernen
bessere Konzepte lernen

Byran 1996

Wie schädlich sind
Fehlvorstellungen?

Normale Begleiterscheinung in kognitiver Entwicklung

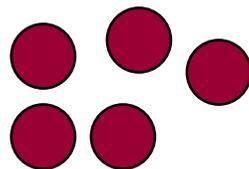
Typische Denkfehler in
Entwicklungsphasen (Piaget)



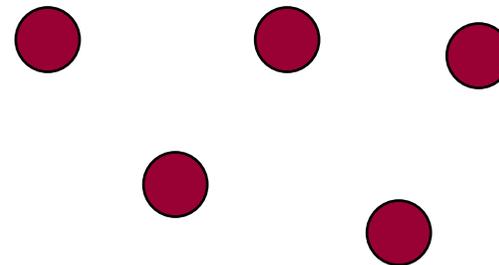
kürzer



länger



weniger

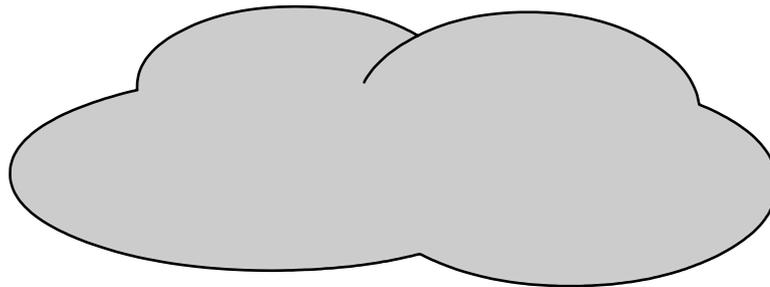


mehr

Objektkonzept der Informatik: Kultivierte Fehlvorstellung

Animismus
präoperationale Stufe (Piaget)

“Warum bewegt sich eine Wolke?”

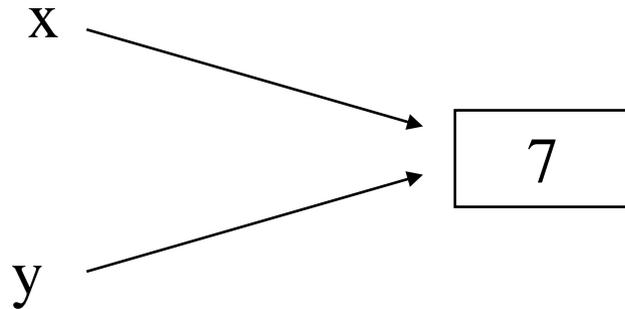


“Sie kriecht über den Himmel wie ein Wurm...”

Konkurrierende Konzepte

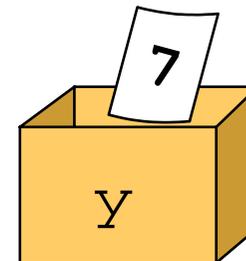
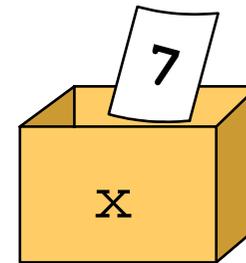
Variable

Name-Objekt-Konzept

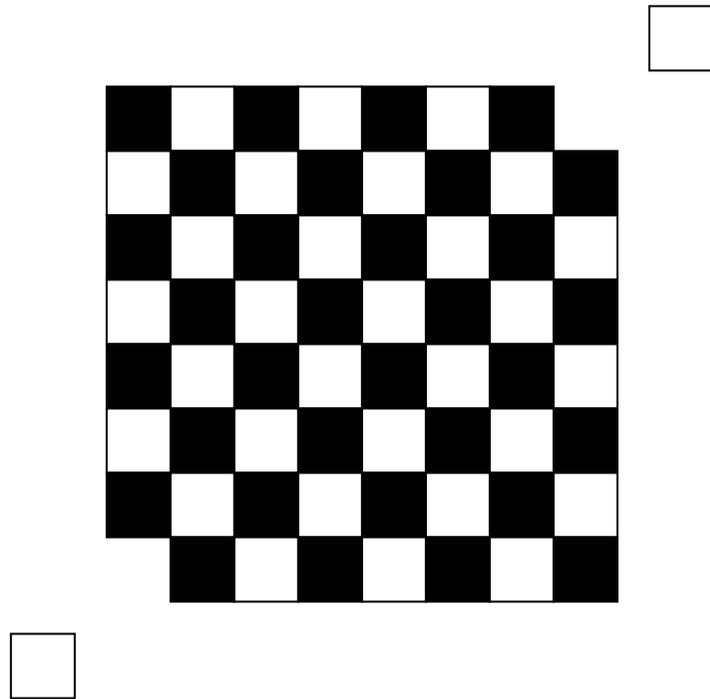


Behälter-Inhalt-Konzept

$x = 7$
 $y = 7$



1.3 Fehlende geeignete Konzepte



Problem:

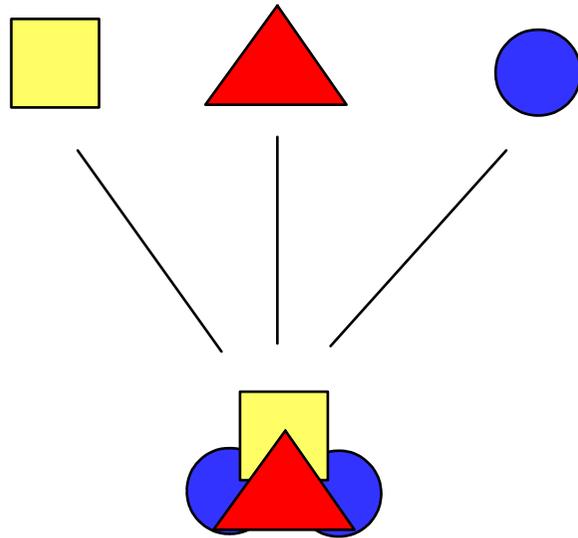
Kann das unvollständige Schachbrett mit 31 Dominosteinen abgedeckt werden?

Geeignete Repräsentation des Problems:

Jeder Dominostein deckt genau ein schwarzes und ein weißes Feld ab.

Anderson 1996

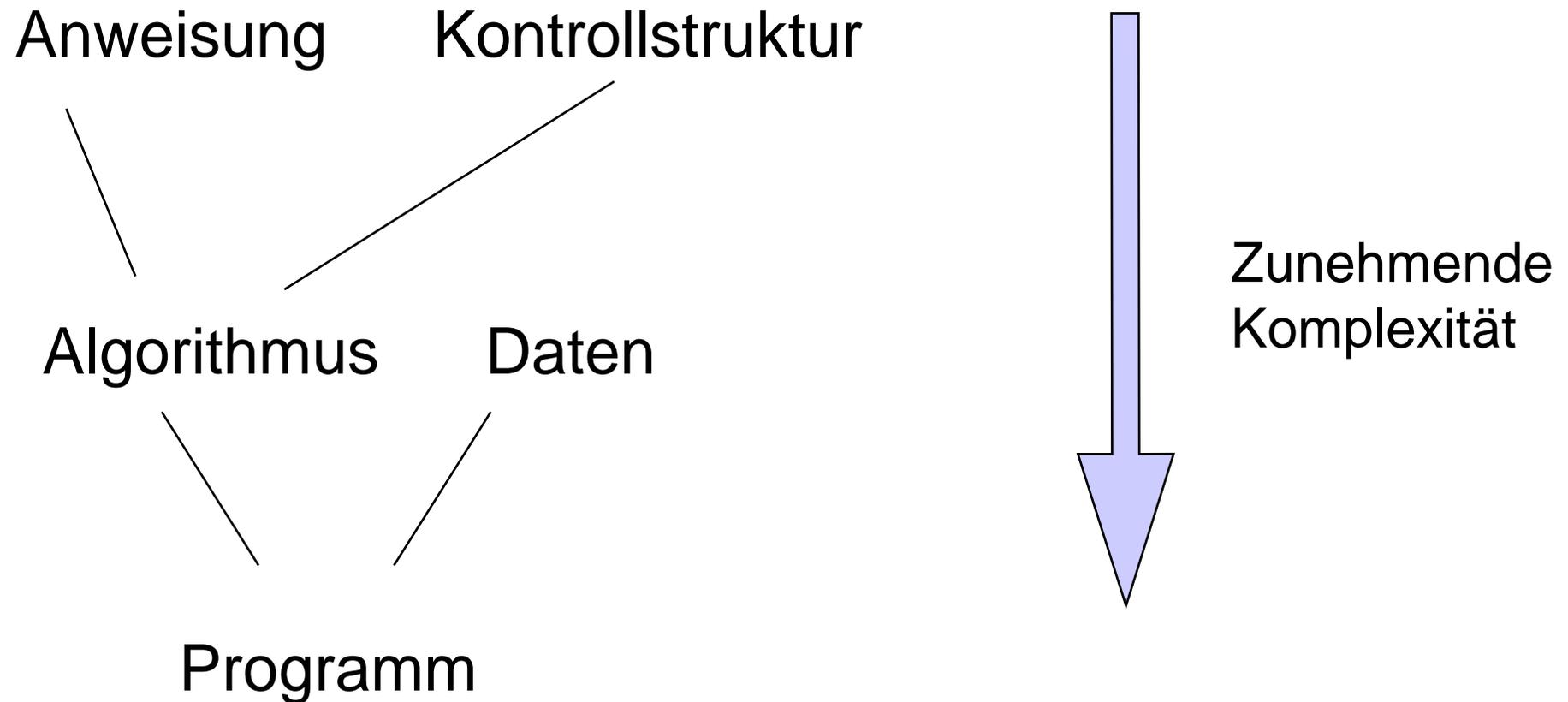
Wissenslücken



Elementare Konzepte

Komplexes Konzept

Modell der sukzessiven Konstruktion



Behinderung des selbstständigen Schließens von Wissenslücken

- keine Zeit

Unterrichtsgespräch: Experten verraten Lösung

- keine Medien

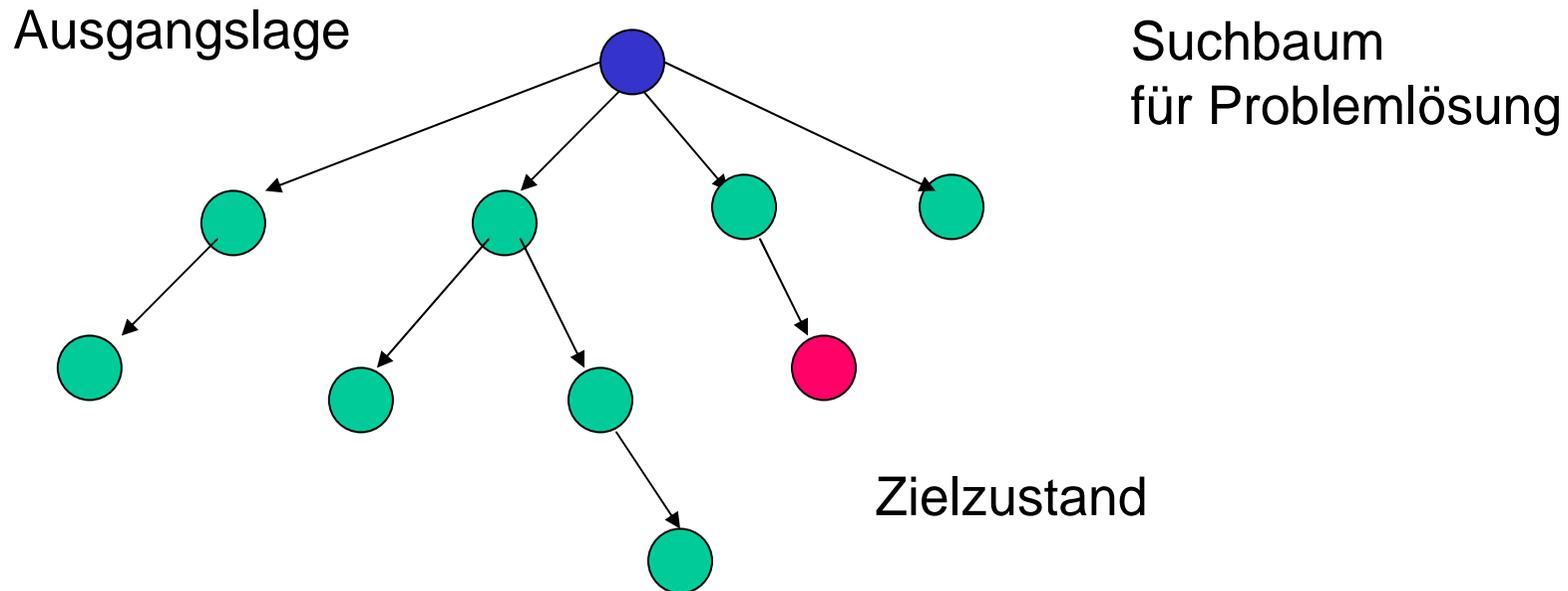
Sprachreferenz, Veranschaulichung

- keine Gelegenheit

Arbeitsteilung im Team, Produktorientierung

- fehlende Lernstrategien

1.4 Fixierung



- Vertikales statt laterales Denken (de Bono)
- Funktionale Fixierung
 - Zwei-Seile-Problem (Maier 1931), Kerzenproblem (Duncker 1966)
- Inkubationseffekte bei Einsichtsproblemen

1.5 Fehlende Lernstrategien

Selbstständiges Lernen hat im Informatikunterricht besondere Bedeutung

- Kognitive Strategien
- Metakognitive Strategien
- Ressourcenbezogene Strategien

Wild/Schiefele 1994

Kognitive Strategien

- **Elaborationsstrategien: Neues Wissen in Wissenstruktur integrieren**

verbale und bildliche Anreicherung, Beispiele finden, Kommentare schreiben, Sprachelemente in Programmierprojekt anwenden

- **Organisationsstrategien: Information in leichter zu verarbeitende Form transformieren**

Stofforganisation, Identifizieren wichtiger Fakten und Argumentationslinien, Geschäftsprozess-Diagramme zeichnen

Weinstein/Meyer 1986

Pintich 1989 ...

Metakognitive Strategien

Kontrolle des Lernprozesses

- Planung von Lernschritten
- Selbstüberwachung des Lernfortschritts
- Regulation des Lernverhaltens

Weinstein 1988

Pintich 1989 ...

Ressourcenbezogene Strategien (Stützstrategien)

Interne Ressourcen

- Anstrengung
- Arbeitszeit
- Aufmerksamkeit und Konzentration

Externe Ressourcen

- Arbeitsplatz
- Literatur
- Arbeitsgruppen

Weinstein 1988, Pintich 1989, Friedrich/Mandl 1992...

Strategie-Inventare

LIST

Inventar zur Erfassung von Lernstrategien im Studium

Wild/Schiefele (Uni Freiburg) 1994

MSLQ

Motivated Strategies for Learning Questionnaire

Pintrich et al. (University of Michigan) 1991

Spezifische Lernstrategien im Informatikunterricht?

Besondere Aktivitäten:

- Zeitkritisches Problemlösen
- just-in-time-Lernen
- Umgang mit Sprachreferenzen
- Experimentieren
- Nutzung von Programmierumgebungen
- Debugging ...

1.6 Fehlende Motivation

Annahmen

- Kognitives Lernen kann nicht erzwungen werden
- Lernen ist aktiver subjektgesteuerter Vorgang

Exkurs: Was ist Motivation?



Verführung? Heimliche Manipulation?

Didaktik: Anknüpfen an vorhandene Motive

Inwiefern nützt der Erwerb des Konzeptes dem Erreichen meiner Ziele?

- Problem in Sinnkontext stellen
- Neugiermotiv
- Verursachungsmotiv

Motive für Informatikunterricht

	Anzahl
1 kreativ sein, ein Produkt schaffen	11
2 Neugierde	3
3 mit anderen zusammenarbeiten	9
4 praktische Lebenshilfe	10
5 sich eine Meinung bilden	2
6 sicherer Job	11
7 wichtig für die Zukunft	10

Befragung von 20 Schülerinnen/Abiturientinnen im August 2000 an der Uni Dortmund. Jede Person konnte drei Motive nennen.

Neugiermotiv

„Gier nach Neuem“

Unbekanntes löst Aktivität aus.

Typisches Neugierverhalten (Oerter):

- **Exploration**
Erkundung der Teile eines Ganzen (z.B. Funktionen einer IDE)
- **Variation des Verhaltens**
Ausprobieren eines Gegenstandes, Entdecken von Verwendungszwecken (z.B. Funktion ausprobieren)

Auslöser für Neugierverhalten

- Objekte oder Ereignisse mit mittlerer Vertrautheit

allzu Fremdes verursacht Angst

- Kognitive Konflikte

Zweifel, Verwirrung, ...

Schlussfolgerung:

Neugierde setzt Wissen voraus

Verursachungsmotiv

Das Gefühl, selbst wirksam zu sein („feeling to be an origin“), bewirkt ...

- hohe Bereitschaft an einer Sache weiter zu arbeiten
- besseres Lernen

Richard DeCharms

1. 7 Kognitive Barrieren in Medien

Ausschluss von Rezipientengruppen
Americans with Disabilities Act (ADA, 1990)
Anlass für barrierefreie Webseiten etc.

Medien im Informatikunterricht

- Sprachreferenzen
- Lehrbücher
- Animationen
- Interaktive Lernmedien
- Mediale Elemente in Interpretern/Compilern

Vermeidung von Barrieren in Lehrtexten

- Redundante Darstellungen aus verschiedenen Perspektiven.
- Anknüpfung an Alltagsvorstellungen
- Bilder, Veranschaulichungen
- Konsistente Begrifflichkeit
- Glossar
- Querverweise
- Beispiele...

Umsetzung kognitiver Strategien
unterstützen

Programmiersprachen

- Redundante Darstellungen aus

2 Erforschung kognitiver Barrieren

Fragen

1 Welche kognitiven Barrieren behindern den Erwerb informatischer Konzepte?

Fehlvorstellungen, Gefühl subjektiver Sinnlosigkeit, schwierige (fehlende) Konzepte

2 Welche Lernstrategien (insbesondere kognitive Strategien) verwenden Schüler beim selbständigen Lernen während eines Programmierprojektes?

Verwendung von Sprachreferenzen, Verwendung von Modellvorstellungen bei der Kommunikation, Experimentieren, Identifikation und Beseitigung von Fehlvorstellungen beim Debugging, ...

Kooperatives Prinzip

Erhebungen in den Unterricht einbeziehen

- Feedback für Schüler

z.B. zur Vorbereitung auf Test oder Klausur

- Teil der Unterrichts-Evaluation

Feedback für Lehrer,

Grundlage für pädagogische Interventionen

- Teil der Unterrichtsdramaturgie

z.B. Befragungen als Einstieg in Diskussionsphase

Nahziel: Materialpakete

- Fertige Lernarrangements mit Erhebungsteil und Lehrerkommentar
- vorgestellt und verteilt bei Python-Lehrerfortbildungen

Erhebungsmethoden

- Analyse von Klausuren und Tests (oder Vortests)
- Interviews
- standardisierte Tests
 - Teil von Leistungstests und Programmieraufgaben
- „Teach back“ mit Videobeobachtung
 - Teil einer mündlichen Prüfung oder „Übungsprüfung“
 - Kolloquium am Ende einer Projektphase
- Monitoring bei Programmentwicklung

2.1 Suche nach Fehlvorstellungen

Verbale Interpretationen

A und B seien Variablennamen.

Was passiert bei einer Zuweisung?

A = B

Kreuze alle zutreffenden Aussagen an.

Der alte Inhalt von A wird zerstört

Der Inhalt von B wandert nach A.

Der Inhalt von B wird nach A kopiert.

Der Inhalt von B wird zerstört.

Das Objekt mit dem Namen B erhält zusätzlich den Namen A

Das Objekt mit dem Namen A erhält zusätzlich den Namen B

Konfrontation mit Analogien

Welcher der folgenden Gegenstände veranschaulicht am besten, was eine Variable ist ?

Eine Schachtel mit einer Aufschrift, in die man einen Zettel geben kann.

Ein Bleistift, mit dem man etwas aufschreiben kann.

Eine CD, auf der Musik gespeichert ist.

Eine Handy-Tastatur, mit der man verschiedene Nummern eingeben kann

Eine Rechenmaschine

2.2 Suche nach kognitiven Strategien

Teach back

in Anlehnung an van der Veer (94)

Verwendung von Modellen bei Erklärungen

Arbeitsweise eines Programms erklären. (5 min)

Vorrat an Symbolen und Abbildungen („Szenokasten“)

Videodokumentation

Beobachtungen: verbale Äußerungen, Zeichnungen,
Zeigen auf Objekte, Arrangieren von Objekten

Selbstbeobachtung bei Programmieraufgabe

Material: Programmieraufgabe und Auszug
aus einer Sprachreferenz mit Beispiel

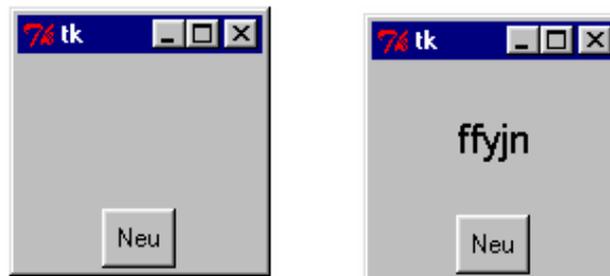
Anschließend Fragen zu kognitiven Strategien

Aufgabe

Für einen Wahrnehmungstest soll ein Text für kurze Zeit auf dem Bildschirm zu sehen sein. Das Anwendungsfenster soll ungefähr so wie in der Abbildung aussehen.

Nach einem Klick auf die Schaltfläche erscheint nach einer Zeit zwischen 1 und 3 Sekunden ein neuer Text aus 5 Zeichen, bleibt 200 Millisekunden lang sichtbar und verschwindet dann wieder.

Entwickeln Sie ein objektorientiertes Programm



Referenzauszug

after()

```
after (ms,func[,arg1[,...]])
```

Die Funktion mit dem Namen *func* wird nach *ms* Millisekunden aufgerufen. Nach dem Funktionsnamen kann man in der Parameterliste die Argumente für diese Funktion aufführen. Zurückgegeben wird eine ID-Nummer, die man zum Abbruch mit `after_cancel()` verwenden kann. Das folgende Beispielskript erzeugt ein weißes Fenster, das nach einer Sekunde grün wird.

```
from Tkinter import *
def neuefarbe(canvas,farbe):
    canvas.configure(bg=farbe)

hauptfenster = Tk()
canvas=Canvas(hauptfenster, height="2.5c",
              width="3.5c", bg="white")
canvas.grid()
canvas.after(1000,neuefarbe,canvas,"green")
hauptfenster.mainloop()
```

Selbstbeobachtung - 1

Kognitive Strategien - Elaboration der Sprachreferenz

- Ich habe die Erklärung sofort verstanden und wusste, wie ich die after()-Methode verwenden muss
- Ich habe das Beispiel studiert
- Ich habe zuerst das Beispiel ausprobiert, bevor ich die Methode in meinem Programm verwendet habe

Selbstbeobachtung - 2

Kognitive Strategien - Programmentwicklung

- Ich habe die after()-Methode mit verschiedenen Parametern ausprobiert
- Ich habe das Programm stückweise aufgebaut. Jedes fertige Stück habe ich solange getestet und verändert, bis es funktionierte.
- Nach dem das Programm fertig war und das leistete, was die Aufgabe verlangte, habe ich es dennoch weiter entwickelt um verschiedene Dinge auszuprobieren.

Selbstbeobachtung - 3

Metakognitive Strategien - Einschätzung des Schwierigkeitsgrads und Lernzuwachses

- Verwendung von Sequenzen
- Herausfinden, welche Methoden die Klasse benötigt
- Definition der Command-Methode für den Button
- Definition des Label-Widgets
- Definition des Button-Widgets
- Dynamische Konfiguration des Label-Widgets

Antwortmöglichkeiten: Noten zwischen 1 und 6

Große Unterschiede deuten auf gute Kontrolle des Lernprozesses₅

Selbstbeobachtung - 4

Ressourcenbezogene Strategien

- Ich habe Experten (Mitschüler, Lehrer) um Hilfe gebeten
- Ich habe Programmtext aus früheren Projekten verwendet
- Ich habe noch in anderen Texten (Sprachreferenz, Materialien aus dem WWW etc.) nachgelesen
- Ich eine gewisse Zeit lang selbst versucht, auftretende Probleme zu lösen, und habe erst dann Experten gefragt, wenn ich in Zeitnot kam
- ...

Antwortmöglichkeiten: 1 bis 6 (1=sehr häufig, 6=nie)

Monitoring der Programmentwicklung

IDE enthält Monitorfunktion, die folgende Dinge registriert:

- Wie oft wurde der Programmtext geändert, bevor das Programm zum ersten Mal fehlerfrei lief?
- Zeitliche Dauer eines Entwicklungsschrittes (Abstand von einer lauffähigen Version zur nächsten)
- Typische Fehler (Problem: Erkennung von Semantik-Fehlern)
- Feedback bzgl. langfristiger Entwicklung (Verbesserung)

3 Methodische Konzepte zur Überwindung kognitiver Barrieren

3.1 Experimente

Beispiel 1

Programmierung

Person A baut etwas aus Legosteinen und beschreibt der Person B ihr Produkt

Person B hört zu (sagt nichts) und baut das Gebilde nach

Beobachtungsaufgabe

Achten Sie darauf, wie Person A den Aufbau des Legosteingebildes erklärt.

Sagt sie, was man *tun* muss?

Oder sagt sie, wie das Gebilde *aussieht*?

Beispiel 2

Variablenkonzept

„Der Inhalt der Variablen mit dem Namen a wird mit dem Wert 3 multipliziert und das Ergebnis in der Variablen mit dem Namen b gespeichert“

Wie lautet die Python-Anweisung?

`b=a*3`

Beispiel 2

Variablenkonzept

„Zum Wert des Objektes mit dem Namen x wird der Wert 10 addiert. Dem Objekt, das die Summe der beiden Werte repräsentiert, wird als neuer Name y zugeordnet.“

Wie lautet die Python-Anweisung?

`y=x+10`

Experimente - Merkmale -

- Mentale Vorstellung mit Beobachtungen vergleichen
- Ungewisser Ausgang
- Neugiermotiv ansprechen
- Sinne einbeziehen

Experimente - Formen -

- Programmieren
 - Evolutionäre Entwicklung,
Debugging
- Rollenspiele
- Arbeiten mit Experimentierumgebungen
- Gedankenexperimente

Experimente - Funktionen -

- Modellvorstellungen überprüfen
- Anregungen für kognitive Strategien
Elaboration von Lernstoff durch Ausprobieren etc.
- Selbsterfahrung

Unterrichtsorganisation, die Experimentieren erlaubt

Projektorientierter Unterricht in Anlehnung an Modelle des Software-Engineering problematisch.

- Starre Vorgehensweise, kein Experimentieren
- Entwurf erfordert viel Erfahrung
- Langwierig
- Arbeitsteilung verhindert Lernerfahrung

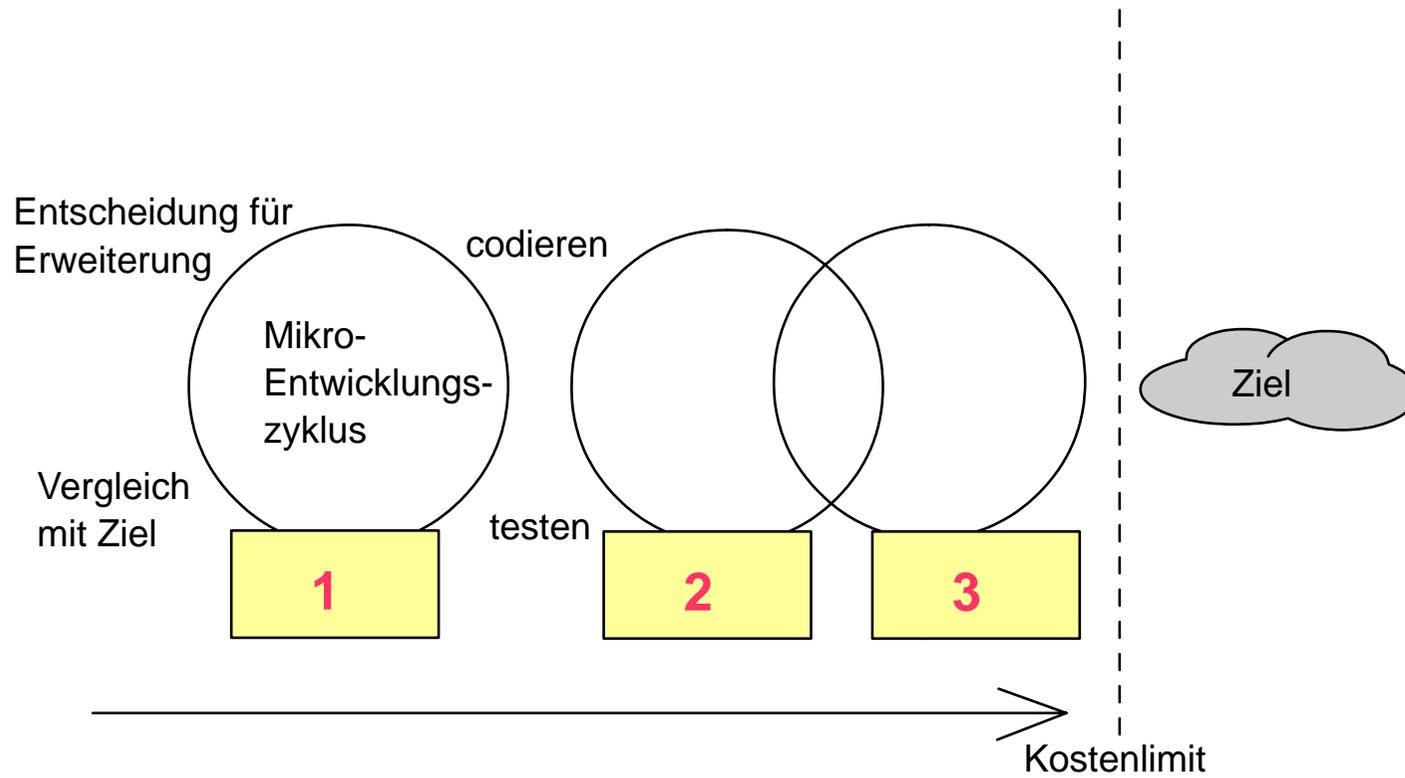
Exkurs: Extreme Programming (XP)

- adaptiv und nicht prädiktiv
- iterativ
- Prozesskontrolle. Nach jeder Iteration:
 - Was haben wir gut gemacht?
 - Was haben wir gelernt?
 - Was können wir besser machen?
- Mensch-orientiert und nicht Prozess-orientiert

XXP: Extremely Extreme Programming

- Fixer Kostenrahmen (z.B. 30 Person-Minuten)
- Zielvorgabe
- Schrittweise Annäherung an das Ziel (Iterationen) ohne Zwang, das Ziel zu erreichen
- Entwicklungszyklen im Minutentakt

XXP



Medien

- Sprachreferenzen
- Interaktive multimediale Animationen
 - Puzzle
 - Experimentierumgebungen
 - Spiele

Was ist zu tun?

- Lernstrategien, mentale Konzepte ermitteln
- Diagnoseinstrumente für Lehrer und Schüler
- Kognitive Barrieren,
Verständnisschwierigkeiten aufspüren
- Methoden zur Verbesserung von
Lernstrategien entwickeln
- Lernarrangements mit neuartigen Medien
designen